IN THE SPECIFICATION:

PLEASE AMEND THE SPECIFICATION AS FOLLOWS:

Please replace the paragraph starting on line 19, page 10 and ending on line 17, page 11 with the following corrected paragraph:

--Servers 102, 103, and 104 are shown operatively coupled via network bus 114 from Client 101 in this client-server network. Any or all of these servers can be remotely located at widely-displaced geographical distances from the client or can be located physically near the client. Also, it is to be understood that a vast number of servers (up to 1000 or more) can be connected on any particular network and only three are shown in this Figure for purposes of enhancing clarity of presentation. Server 102 is operatively coupled to disk array or storage device 106; server 103 is operatively coupled to disk arrays or storage devices 107 and 108 as well as to network cloud 115 representing Network Attached Storage (NAS) or Storage Area Network (SAN) environments and designated herein SAN/NAS or NAS/SAN for convenience purposes; and, server 104 is operatively coupled to storage device 109. Server templates 111 are shown associated with server 102 and are stored in storage device 106; server templates 112 are shown associated with server 103 and are stored in storage devices 107 and/or 108 as well as within storage devices (not shown) operating within or associated with NAS/SAN cloud 115; and, server templates 113 are shown associated with server 104 and are stored in storage device 109. These server templates are also software objects and they have been deployed or assigned by client 101 to databases in their respective server-based storage devices. Host or client 101 where the configuration UI software is running maintains a

database on its storage device 105 of all server templates used throughout the site or network (in this instance templates 111, 112, and 113). These server templates as well as the client templates are thus also under control of client 101 and will also be discussed in more detail in connection with other figures hereinbelow.—

Please replace the paragraph starting on line 8, page 18 and ending on line 13, page 19 with the following corrected paragraph:

--The foregoing algorithms and discussion thereof set forth essentials of operation of an events-notification scheme with which certain embodiments of the present invention, involving template-calibration or template-synchronization or template-ambiguity-purging, are particularly useful. These embodiments ensure against multiple templates with the same name having different contents, which, if not corrected are erroneous conditions which can negatively impact operation of the events-notification scheme. Referring to Fig. 5, a flowchart depicts an algorithm of an aspect of the present invention associated with purging ambiguity in a situation where pre-existing server template-object content has a certain name and different pre-existing or newly-retrieved client template-object content has the same certain name. The algorithm starts with step 501 where the head-end station (client) retrieves a particular pre-existing template from a particular database maintained on a particular storage device associated with a particular server-host, such as, for example, template 113 from a database maintained on storage device 109 associated with host 104 in Fig. 1. (Such particular template could have been located on a storage device associated with a host (not shown) within or related to cloud 115 of Fig 1.) After retrieval is accomplished, the algorithmic process moves to decision

block 502 wherein a comparison is made between the name of this retrieved pre-existing server template and the name of every template in this head-end station database. If the name is not the same the algorithmic process moves to block 507 wherein the retrieved template is added to this head-end database. Block 507 is returned to the start of block 501 where the [[tnext]] next template is retrieved. But, if the name of the retrieved pre-existing template was the same as, or identical to, the name of any template in the head-end station then the algorithmic process would have moved to decision block 503. In block 503, a decision block, the two identically-named templates' contents are compared and if the contents are not identical the process moves to block 508. In block 508 the conflict is resolved and the process moves from there back to the beginning at block 501. The input to and output from block 508 are tabbed "A" and "B" respectively to coordinate with Fig. 6 which is a subroutine comprising block 508 and about which more discussion will take place hereinbelow.—

Please replace the paragraph starting on line 1, page 22 and ending on line 7, page 23 with the following corrected paragraph:

--Finally, there could be a network scenario where there are separate networks each having their own segregated clients and servers, but where one or more servers from one or more other networks, for some reason, have been arranged to interact with the above-noted client of this network. Referring to Fig. 7, this scenario is presented in a block diagram. The above-noted client C1 701 (equivalent to Client 101 of Fig. 1) is shown operatively networked to server-database combinations or server-locations 703, 704, and 705 (for purposes of simplification, these three singular blocks should be

viewed as combining the functions of server, database, and template as represented separately in Fig. 1). Components 701, [[702, 703]] 703, 704, and [[704]] 705 comprise network I, as shown, which is separated from network II by imaginary demarcation line 709. (Many more networks which could have been shown with many servers per network are not shown to enhance clarity of presentation.) Network II comprises different client C2 702 which is operatively coupled to its server-database combinations 706, 707, and 708 (to be viewed similarly to 703, 704, and 705). The cross-network connection 710 shows that above-noted client 701 is operatively coupled out of its normal network to server [[706]] 708. In this instance if there is a conflict between names and contents of templates stored on databases associated with server [[706]] 708 and client 701, client 701 updates conflicting templates stored on databases in server [[706]] 708 to conform to its own templates. In such a case, such updated templates in server [[706]] 708 are also changed relative to expectations of other client C2 702. However, an assumption is made that Client C1 is operating with new data and Client C2 is operating with old or obsolete data. In other words, the most current event-notification-establishment-user can over-ride template names or contents for templates stored in databases of servers that are common to itself and other clients. Thus if client C2 702 had established its event-notification parameters earlier than the establishing of event-notification parameters for client C1 701, then any client C1 701 templates imposed on databases associated with server 706 in Network II shall be further imposed on Client C2 702 as being the "latest" or "most preferred" event response or template. This is summarized in decision block 605 of Fig. 6 as an action to resolve conflict by

"update local template", which means, in terms of the example and scenario used herein, to update the local template in client C2 702.—